DOCUMENT RESUME

ED 199 903 EA 013 409

AUTHOR Friedman, Charles P.
TITLE Software Organization in Student Data Banks for Research and Evaluation: Four Institutional Models.
PUB DATE Apr 80
NOTE 20p.; Paper presented at the Annual Meeting of the American Educational Research Association (Boston, MA, April 7-11, 1980).

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Programs; Databases; *Data Processing; Educational Research; Higher Education; Institutional Research; *Program Design; Research Methodology; Student Research

ABSTRACT

Student data banks for ongoing research and evaluation have been implemented by a number of professional schools. Institutions selecting software designs for the establishment of such systems are often faced with making their choice before all the possible uses of the system are determined. Making software design decisions involves "rational" factors such as considering available approaches, anticipating the uses of the system, determining an access policy, and evaluating available hardware and software resources. "Nonrational" factors involved include time deadlines, the familiarity of the users with systems or packages, and institutional priorities for utilization of existing facilities. Characteristics of software designs include the type of managerial package, the data building blocks employed, the method of relating these blocks to each other, search methods available for use, the accessibility of data for updating and editing, and features addressing the designs' built-in limitations. This document, intended for those familiar with computers but not for computer professionals, discusses four systems currently in operation in schools of medicine. The intent of the discussion is to illuminate differences in software design characterstics and to explore the operating implications of these differences. (Author/PGD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED199903

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Software Organization in Student Data Banks for Research and Evaluation: Four Institutional Models

Charles P. Friedman
Office of Medical Studies
University of North Carolina
School of Medicine
322 MacNider Bldg. (202H)
Chapel Hill, NC 27514

(This paper was presented in a symposium at the 1980 meeting of the American Educational Research Association)

Running head: Software Organization

EA 013 409

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
C. P. Friedman
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ABSTRACT

Student data banks for ongoing research and evaluation have been implemented by a number of professional schools. An institution wishing to establish such a system must choose among a virtually infinite set of alternatives for a software design. They often must make such a choice in an uncertain environment since the exact nature of system use cannot be predicted. To facilitate software design decisions, this paper identifies numerous "rational" and "non-rational" factors that influence these decisions, and a set of characteristics descriptive of the software designs themselves. Using as examples four systems currently in operation in schools of medicine, the narrative examines ways in which software designs differ on the characteristics and some of the operating implications of these differences.

INTRODUCTION

At institutions of professional education, personnel wishing to install a computerized data bank for research and evaluation find themselves in an interesting dilemma. At the formative stages of such a system, they must select or devise a structure for the data files they plan to create; however, they often must do this based on partial information about how the new system is to be used. Data banks often justify their creation through arguments of flexibility--ability to respond to a diversity of needs and requests--yet decisions about system structure that permanently constrain the responsiveness of the system must be made before the precise nature of actual requested uses can be known. Designers of such systems can only make educated guesses, recognizing that no solution to the structure problem will be perfect. One way of making such a guess more educated is to clarify what others in the field have done to respond to their own perceptions of needs, usage patterns, and organizational context. To foster such clarification is the purpose of this paper.

We will examine the approaches to data bank organization by comparing and contrasting the approaches of four medical schools with systems up and running: the University of Arizona (D. Anderson, 1979), Jefferson Medical College;[1] McMaster University (Woodward, 1977), the University of North Carolina (Cheatham et al., 1979). The tone of this paper will be descriptive and non-judgmental, as all four systems appear to be serving very adequately the purposes for which they were designed. Differences between the approaches

are striking, so that a comparative approach can bring into full focus the compromises indigenous to the software design process.

Before setting out, it may be useful to expand upon the limitations of this paper. First, the work is designed for an audience acquainted with computers and computing, but not an audience of programmers and computer scientists. This limitation is imposed recognizing that programmers and computer scientists typically constitute only part of the group making final system design decisions, and the group least in need of the analysis this paper will provide. The second limitation restricts the systems under analysis to those intended for educational or institutional research and evaluation, excluding systems designed in whole or part to execute administrative functions such as record-keeping or accounting. Lastly, we will not emphasize here the variables that comprise the content of individual banks, but rather the structure of whatever content about students is selected for inclusion. Although the two decisions are not fully independent, we make that assumption to clarify the analysis.

WHAT DICTATES A SOFTWARE DESIGN?

It is often useful to divide these factors into two groups: first the "rational" factors, meaning those considerations that facilitate application of a "rational" approach to planning and decision making (Allison, 1971), whereby one examines all existing or projected alternatives and then selects that which maximizes projected benefits while minimizing resources expended to attain these benefits. Second, we list the (non-rational) factors, always operative to some degree,

that mitigate against application of a pure rational approach.

Rational factors include:

1. Available approaches: What are the extant alternatives and how have they worked in practice at other institutions?
2. The anticipated use of the system: Data bases emphasizing research and evaluation may be used for many purposes. These range from ten-year longitudinal research projects interrelating scores of variables to applied curriculum evaluation studies that might produce the distribution of grades or test scores in specific courses in specific years. Software structures vary in the convenience and speed with which they perform these different types of tasks.
3. Access policy: Who will use the system in the sense of actually retrieving, updating and analyzing data? If personnel without formal training or experience will interact directly with the system, one must incorporate plain language interactive features; if direct access is to be restricted to a particular experienced group, then use of programming codes and other shortcuts is permissible, but turnover in personnel might require lengthy training of replacement staff.
4. Available hardware and software resources: The number, type, location, availability, and usage cost of computers and input-output devices to these computers is a major factor. A related factor is the software such as file management and statistical packages available in

conjunction with the hardware. System designers addressing these factors must not only assess the present situation; they must also look into the future and make their designs maximally compatible with what will be available in five years.

While designers of systems strive for the rational planning ideal--and, indeed, a self-conscious goal of this paper is to parsimoniously array the extant alternatives and thereby promote rational planning--none attain it in practice. An additional set of factors operates to constrain design decisions. These factors are termed "non-rational;" however, they should not be considered irrational as they are a reality of organizational life. Factors that often interfere with pure application of rational planning are:

1. Time deadlines: System designers often promise a working result by a specified date. A less than optimal system from a pure cost-benefit perspective may be selected because it makes possible attainment of the deadline. Many data systems are supported by grants or internal discretionary funds; this factor make deadline attainment even more important to gain credibility necessary for continuing support.
2. Familiarity: Many software alternatives are based upon availability of packages such as SPSS--the Statistical Package for the Social Sciences (Nie et al., 1975) and SAS--the Statistical Analysis System (SAS Institute, 1979). If one is already experienced in use of a particular package, one is naturally inclined toward employing it.

3. Institutional Priorities: Within institutions there may exist pressures to increase utilization of some facilities or hardware installations, even if those facilities cannot accommodate the most desirable software design.

CHARACTERISTICS OF SOFTWARE DESIGNS

Individuals choosing software designs respond to the rational and nonrational features cited above, but they also organize their thinking by looking at several characteristics inherent in software designs themselves. The major challenge in designing a data bank derives from the inevitable variability in student progress and performance within a curriculum; and possible changes over time in the information content stored about each cohort of students, or changes in coding of the information. Many of the six interactive features of data bank designs, listed below, reflect conscious attempts to respond elegantly to this challenge; so that the computer keeps track of these variations and, when executing statistical analyses, deals with them efficiently and automatically. It should be emphasized that these six features interrelate in the sense that changing any one of them, for a given system, would inevitably change the other features as well. Table 1 summarizes how each of the four systems addresses the six characteristics.

Insert Table 1 About Here

---

1. Managerial Package: Most data bases employ a commercially available software package that, in effect, organizes the data files comprising the entire system. Use of a managerial package is not mandatory, as the information in the system can simply be fed as "raw" data to any computer program for statistical analysis. Also, choice of a particular package does not fully determine the design of a system; the packages themselves are very flexible and many decisions must still be made by the system designers. For file management both UNC and Jefferson employ SAS, the Statistical Analysis System (SAS Institute, 1979); while McMaster uses SIR, the Scientific Information Retrieval System (Anderson et al., 1978). Both packages store data as "save files" connecting economically to programs that perform statistical computations. In addition to SAS, Jefferson's system employs for some purposes a specialized data base management system provided by a local vendor.

2. Data Building Blocks: A major design decision that must be made, whether one uses a managerial package or not, relates to data format and organization. Usually the variables stored for each case (each student) are grouped in some way to facilitate processing, by enabling the system user to perform operations on an entire group of variables rather than one variable at a time. These aggregations of variables are the building blocks of a system. The Arizona system groups its data into 80 character "card images." UNC and McMaster group data in logical units; e.g., pre-admission data or performance in curricular year 1. The Jefferson system does not group data beyond the level of individual variables.

3. Relations Among Building Blocks: After the variables are grouped, they must be arrayed in some manner so the computer in effect knows how to assemble the values of those variables for each student. In the Arizona system, the "card images" pertinent to each individual student are joined in a long string; the same relationship applies in the Jefferson system, where all variables pertinent to each student are arrayed in a string. In both of these systems, then, the data assume

the form of a rectangular matrix. In the McMaster system groups of variables are interrelated in a hierarchical pattern, instead of a string, such that some groups are subsumed by others. Thus in the McMaster system, the organization of the data in the computer conforms closely to the logical structure of the data in the real world. This simplifies many retrievals from their system. In the UNC system, data about any particular student are scattered throughout sub-files. Each sub-file contains data about a group of students taking a particular curriculum year in the same academic calendar year. A separate "control file" remembers each student's progress through the curriculum, and knows, thereby, in which set of sub-files to find all the information about each student.

4. Search Methods: Almost all uses of data bases require the searching of files for specific individuals or groups of individuals having specific characteristics. Once these subgroups are selected, statistical analysis can be performed using variables of interest. Searching for particular individuals or groups can be strictly sequential where the computer mechanically examines each case in the file, in the order the cases happen to appear, and picks a case out if it has the desired characteristic(s). By contrast, other systems have an index sequential search capability where the cases can be sorted according to a "key" variable; the computer then capitalizes on the

known ordering of the sorted cases to search the files more quickly. The Jefferson and McMaster systems both have capability for index sequential searching.

5. Update and Edit Procedures: All systems must have facilities for editing files and for updating them as students advance through a curriculum and generate more data. Some managerial packages have built-in routines for updating or editing information in the file; others require programs to be written locally. Whether built-in or locally produced, these routines can operate either interactively or through batch processing. Massive updates adding large amounts of data generally operate most efficiently when processed en masse (in batch) while minor modifications of individual student records are most easily done interactively. The Jefferson and McMaster systems can update and edit either in batch or interactive mode; Arizona and UNC employ batch mode only.

6. Economizers: Every system design has limitations, and most data banks have features that address their limitations in some systematic way, compensating to the maximum possible extent. The Jefferson and Arizona systems, with their rectangular data matrix structure, are not particularly conservative of computer storage space. This is because space must be allocated in every student's record for all variables

ever stored about any student. The result is many blank "place holders" within the file. To compensate, the Arizona system uses a custom-made program that, in effect, generates many embedded place holders when needed for certain retrievals. The main data files do not, in actuality, contain as many embedded blanks as would otherwise be required. The Jefferson system has two distinct files: their main file has all required embedded blanks and a smaller file (based on SAS) contains only the variables frequently used in retrivals or analyses. The smaller file, as a function of its thoughtfully selected contents, has variables that have been assessed for almost all students and therefore requires fewer blank place holders. The UNC system has begun to store often-requested information in the "control file" that originated only to remember each student's progress over time through the curriculum. In this way, many retrievals can be made using the control file only, without requiring access to the large and more complex files containing the balance of the data.

OBSERVATIONS ON THE FOUR MODELS

The following analysis addresses in turn each of the "rational" criteria for system design, presented earlier, and follows directly from the system characteristics embodied in Table 1.

1. Approaches available at the time of system creation: It is evident that the Arizona system was designed (in 1973) before the more sophisticated file management capabilities inherent in SAS and SIR were widely available.[2] The Arizona system's simplicity of design, with some attendant difficulties in actual use, must be in part attributable to that item of history. The Jefferson system, too, was established earlier than those at UNC and McMaster; and its structure--using SAS in a secondary file--likely follows to some degree therefrom.
2. Anticipated use of system: The file structures in all cases appear strongly related to the original goals in setting up the system. The Arizona and Jefferson systems were initially envisioned as bases of longitudinal studies of medical students. In the Arizona case, the initial goal was to evaluate an innovative three-year curriculum implemented at a new medical school. Acknowledging that these two systems have since proved very useful in more varied ways, their less elaborate structure may have derived in part from their more circumscribed initial missions. By contrast, the UNC and McMaster systems were envisioned from the outset as standing resources for institutional research

and curriculum evaluation. This direct connection with the process of administering the school by informing day-to-day decision-making gave rise to a system design necessarily more complex to accommodate a wider variety of anticipated types of use.

The UNC system apparently also forecast differently from the other schools the types of analyses it would be required to perform. As a result, the other three system designs are engineered to easily retrieve complete records of individual students, since all information about each student is contiguous in the computer memory. By contrast, the UNC system is structured about discrete points in the medical curriculum. From the UNC file structure it is simple to retrieve information about cohorts of students who entered medical school or took Year 3 of the curriculum at the same time; while some manipulation is needed to assemble all available information about one individual student.

3. Access policy: The less complex file structures (Arizona and Jefferson) have some built-in inefficiencies pertaining to information storage and retrieval. They are, however, the easiest systems for computer laymen to understand and are the most flexible in terms of ability to interface directly with many other software packages. At the expense of some convenience of use, both Arizona and Jefferson have developed mechanisms to realize additional operating economy by eliminating gaps in the data files.

Conversely, through some programming, the data from the more complex (UNC and McMaster) systems can be transformed into a simpler structure, should that become necessary. The SIR system employed by McMaster employs the syntax of SPSS, familiar to many social and behavioral scientists. This potentially provides direct system access to a larger number of people. Jefferson and McMaster have built in the capability for plain language interactive update, making this a clerical task when appropriate, while at UNC and Arizona, this remains exclusively a "batch" task for a programmer.

4. Available hardware and software: Three of the four systems have made commitments to statistically-oriented data management packages: SIR in the case of McMaster; SAS in the case of Jefferson and UNC. However, while UNC has made a total commitment to SAS, Jefferson's main file is statistical software independent. While UNC, Arizona, and McMaster are contiguous with major comprehensive universities having their own large computer centers, Jefferson is not and uses a private vendor for hardware access. Jefferson's system design therefore had to recognize the possibility of changing hardware arrangements, while the other three institutions were likely precluded from seeking arrangements off-campus. The SIR system, despite its many elegant features, is not widely available to date in university computer centers.

This paper will not attempt to trace the particular ways in which the "non-rational" factors affected design decisions. This author

will, for his part, acknowledge that all three factors cited earlier were operative at various times in the design of the UNC system. Rather than apologizing for these influences, it is better to acknowledge their inevitability and then go through the useful exercise of trying to understand what their effects have been. Sometimes, the effects of non-rational planning methods are serendipitous.

ACKNOWLEDGEMENT

The author wishes to thank David Anderson, Gary Anderson, Carol Porter, and John Veloski for their personal advice and for providing information essential to preparation of this manuscript.

NOTES

[1]Design details of the Jefferson system were conveyed via personal communication between the author and Mr. John Veloski.

[2]The first comprehensive SAS user's manual became available in 1972.

REFERENCES

Allison, G.T. (1971) "The essence of decision: Explaining the Cuban missile crisis." Boston: Little Brown.

Anderson, D.O. (1979) "Arizona longitudinal data base." University of Arizona: Unpublished manuscript.

Anderson, G.D., Cohen, E., Gazdzik, W. and B. Robinson. (1978) "SIR, a working system for managing large and complex research data." Paper presented at Computer Science and Statistics: 11th Annual Symposium on the Interface, Raleigh, NC.

Cheatham, C.M., Porter, C.Q. and C.P. Friedman. (1979) "Developing a student data base for institutional research and evaluation." Paper presented at the Research in Medical Education Conference, Washington, November 1979.

Nie, N.H., Hull, C.H., Jenkins, J.G., Steinbrenner, K. and D.H. Bent. [eds.] (1975) Statistical Package for the Social Sciences (2nd ed.) New York: McGraw-Hill.

SAS Institute (1979 ed.) SAS Users Guide. Raleigh, NC.

Woodward, C.A. (1977) "Proposal for an educational data bank within the faculty of health sciences. Initial focus: undergraduate M.D. programs." MacMaster University: Unpublished manuscript.

Table 1

Characteristics of the Four Systems

	Managerial Package	Data Building Blocks for Each Case	Relations Among Building Blocks
Arizona	None	A Card Image	String
Jefferson	Local DBM Package and SAS	A Variable	String
McMaster	SIR	A Record: Logical Group of Variables	Hierarchy
UNC	SAS	A Logical Group of Variables	Control File Remembers Where to Find All Data About Each Student

	Search Method	Update & Edit	Economizers
Arizona	Sequential Only	Batch Only	Program to Generate Place Holders When Needed
Jefferson	Index Sequential or Sequential	Batch or Interactive	"Secondary" File That Stores Only Popular Variables
McMaster	Index Sequential or Sequential	Batch or Interactive	Many Economizers Built Into SIR System
UNC	Sequential Only	Batch Only	Including Extra Data in Control File